(a) selecting an XML encoded test case for execution on a host machine using a web-interface provided by an application server, wherein said test case comprises one or more GUI object test steps;

(b) selecting the host machine on which to run said test case using said web-interface provided by said application server, said host machine comprising an automation tool and a whole or partial copy of the computer readable code of the ~~software~~ GUI being tested;

(c) encoding a test case file comprising the one or more GUI object test steps of said selected test case and the name of an automation tool GUI environment file;

(d) transmitting said XML encoded test case file from said application server to said selected host machine;

(e) receiving said XML encoded test case file and decoding the contents thereof; wherein said decoding comprises generating a test script by parsing GUI object functionality ~~test step~~ syntax recognized by said automation tool from said XML encoded test case file and receiving a copy of the automation tool GUI environment file;

(f) loading said test script, said automation tool GUI environment file, and said whole or partial copy of the computer readable code of the ~~software~~ GUI being tested into said automation tool and executing said loaded test script using said automation tool thereby testing ~~a software application~~ said GUI.

Claim 2. *(Canceled)* ~~The method according to claim 1 further comprising transmitting status information about said test case from said host machine to said application server and providing said status information to a user by way of said web interface.~~

Claim 3. *(Canceled)* ~~The method according to claim 2 further comprising transmitting status information about one or more test steps of said test case from said~~

~~host machine to said application server and providing said status information to a user by way of said web interface.~~

Claim 4. *(Canceled)* ~~The method according to claim 1 wherein said test case file is selected from a flat file, markup language encoded file, XML file, HTML file, ASCII file, or XHTML file.~~

Claim 5. *(Original)* The method according to claim 1 further comprising scheduling the execution of said selected test case on said selected host machine, wherein said scheduling comprises a time and a date for executing said selected test case on said selected host machine.

Claim 6. *(Original)* The method according to claim Claim 1 further comprising a querying of the application server by the host machine whether a test case has been assigned to the host machine.

Claim 7. *(Original)* The method according to claim 6 further comprising said host machine requesting from the application server the encoding and transmitting of a test case file based on the test case assigned to said host machine.

Claim 8. *(Once Amended)* The method according to claim 1 further comprising authoring a test case using said web interface and encoding said test case into XML format, wherein said authoring comprises adding GUI object functionality steps or modifying existing GUI object functionality steps by selecting from a plurality of GUI objects representing GUI object functionalities.

Claim 9. *(Once Amended)* The method according to claim 8 ~~further comprising modifying~~ wherein said GUI object functionalities are selected from a group consisting essentially of an action, procedure, and expected result.

Claim 10. *(Canceled)* ~~The method according to claim 8 wherein said GUI objects are preset GUI objects.~~

Claim 11. *(Original)* The method according to claim 8 wherein said GUI objects are manually created and an action, procedure, or expected result corresponding to said manually created GUI object is entered using automation tool recognized syntax.

Claim 12. *(Once Amended)* A computer system for web-interactive ~~software~~ GUI testing comprising:

(a) an application server suitable as a web server operatively coupled to a database wherein said application server comprises a computer readable storage medium having computer readable code means for providing a web interface to one or more users, computer readable code means for populating said web-interface with data gathered from said database, and computer readable code means for populating said web-interface with data received from one or more host machines,

(b) one or more user machines in communication with said application server ~~suitable~~, said user machines comprising computer readable storage media including computer readable code means for interacting with the web interface provided by said application server, and

(c) one or more host machines in communication with said application server, said one or more host machines comprising computer readable storage media including an automation tool, a full or partial copy of the computer readable program code of a ~~software application~~ GUI to be tested, and computer readable

code means for loading an XML encoded test case and parsing said test case into a test script, an automation tool GUI environment file, and the computer readable program code of said software application into said automation tool and ~~commanding the execution of~~ executing said test script by said automation tool.

Claim 13. *(Once Amended)* The computer system according to claim 12 wherein said application server further comprises:

(a) computer readable program code means for enabling a user to select ~~of~~ an XML encoded test case,

(b) computer readable program code means for enabling a user to select a host machine on which to execute said selected test case,

(c) computer readable program code means for XML encoding a test case file wherein said test case file comprises one or more ~~of~~ GUI object test steps ~~of said selected test case~~ and the name of an automation tool GUI environment file;

(d) computer readable program code means for transmitting said XML encoded test case file to a host machine.

Claim 14. *(Canceled)* ~~The computer system according to claim 13 wherein said encoded test case file is selected from a flat file, markup language encoded file, XML file, HTML file, ASCII file, or XHTML file.~~

Claim 15. *(Original)* The computer system according to claim 13 wherein the application server further comprises computer readable program code means for scheduling the execution of said selected test case on said selected host machine, wherein said scheduling comprises a time and a date for executing said selected test case on said selected host machine.

Claim 16. *(Once Amended)* The computer system according to claim 13 wherein the application server further comprises computer readable code means for enabling a user to author ~~of a~~ an XML encoded test case using said web interface wherein said user adds GUI test steps or modifies existing GUI test steps by selecting from a plurality of GUI objects.

Claim 17. *(Original)* The computer system according to claim 16 Claim 16wherein said application server further comprises computer readable code means for enabling a user to modify an action, procedure, or expected result corresponding to a selected GUI object.

Claim 18. *(Canceled)* ~~The computer system according to claim Claim 16 wherein said GUI objects are preset GUI objects.~~

Claim 19. *(Once Amended)* The computer system according to claim 12 Claim 12wherein said one or more host machines further comprise

(a) computer readable program code means for querying said application server whether a test case has been scheduled for execution by said host machine,

(b) computer readable program code means for requesting the encoding into XML ~~of~~ and the transmission of a test case file by the application server,

(c) computer readable program code means for receiving said XML encoded test case file,

(d) computer readable program code means for decoding said XML encoded test case file, said decoding comprising generating a test script by parsing automation tool recognized syntax from said XML encoded test case file, and resolving the name of an automation tool GUI environment file from said test case file,

(e) computer readable program code means for requesting the transmission of and receiving said automation tool GUI environment file from the application server,

(f) computer readable program code means for updating the application server with the status of a test case assigned to be executed on said host machine or the status of the one or more GUI object test steps included in said test case.

Claim 20. *(Once Amended)* A method for managing ~~software~~ GUI testing requirements comprising:

(a) providing a web interface wherein one or more users can create one or more requirements folders using said web interface and saving said one or more requirements folders in a relational database coupled to an application server that provides said web interface,

(b) providing a web interface wherein one or more users can create one or more test cases using a web interactive authoring tool and saving said one or more test cases in a relational database coupled to an application server that provides said web interface, wherein said test cases comprise one or more GUI object test steps, and

(c) providing a web interface wherein one or more users can sort said one or more test cases within said one or more requirements folders~~.~~; and

(d) providing a means for encoding said one or more test cases into XML format.

Claim 21. *(Original)* The method according to claim 20 further comprising providing a web interface wherein one or more users can create one or more sub-requirements folders using said web interface and saving said one or more sub-requirements folders in said one or more requirements folders; wherein both said requirements and said sub-

requirements folders are saved in a relational database coupled to said application server that provides said web interface.

Claim 22. *(Once Amended)* The computer system according to claim 12 wherein said application server further comprises

(a) computer readable program code means for providing a web interface wherein one or more users can create one or more requirements folders using said web interface and saving said one or more requirements folders in a relational database coupled to an application server that provides said web interface,

(b) computer readable program code means for providing a web-interactive interface wherein one or more users can create one or more XML encoded test cases using a web-interactive authoring tool and saving said one or more XML encoded test cases in a relational database coupled to said application server that provides said web interface, and

(c) computer readable program code means for providing a web interface wherein one or more users can sort said one or more XML encoded test cases within said one or more requirements folders.

REMARKS

This responds to the Office Action mailed on May 18, 2006, and the references cited therewith.

Claims 2-4, 10, 14 and 18 are canceled. As a result, the claims originally numbered as 1, 5-9, 11-13, 15-17 and 19-22 are now pending in this application.

Claims 1, 8, 9, 12, 13, 16, 19, 20 and 22 are amended. In particular, the claim language has been amended to relate particularly to the testing of Graphical User Interfaces (GUI) and their corresponding Graphical User Interface objects by incorporating test steps that test the functionality of such GUI objects. Furthermore, the

claims have been amended to incorporate a limitation originally recited in claims 4 and 14 into the independent claims to recite XML encoded test cases.

Applicant submits that no new matter has been added to the specification or claims by amendment.

Objections to the Drawings

The Examiner's objections to the drawings are acknowledged. Enclosed herewith, Applicants submit replacement drawings prepared by a professional draftsperson. Applicants aver that no new matter has been added by these replacement drawings and respectfully requests withdrawal of the objection to the drawings.

Objections to the Priority Language

The Examiner's objection to the priority language in the specification is acknowledged. Enclosed herewith, Applicants submit a replacement page correcting the term "USSN," to "U.S. Provisional Application Serial Number." Applicants respectfully request withdrawal of this objection.

Objections to the Abstract

The Examiner's objection to language in the abstract is acknowledged. Enclosed herewith, Applicants submit a replacement page with corrections. The "and/or" language objected to by the Examiner has been replaced with "or." Applicants respectfully request withdrawal of this objection.

§102 Rejection of the Claims

Claims 1-19 were rejected under 35 U.S.C. § 102(b) as allegedly anticipated by Glenn et al. The Examiner alleges that Glenn et al. disclose each element of the claims.

A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference. *Verdegaal Bros. v. Union Oil of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir.

1987). The identical invention must be shown in as complete detail as is contained in the claim. *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ 2d 1913, 1920 (Fed. Cir. 1989). To constitute anticipation, the claimed subject matter must be identically disclosed in the prior art. *In re Arkley*, 172 U.S.P.Q. 524 at 526 (C.C.P.A. 1972). For anticipation, there must be no difference between the claimed invention and the reference disclosure, as viewed by a person of ordinary skill in the art. *Scripps Clinic & Res. Found. v. Genentech, Inc.*, 927 F.2d 1565, 18 USPQ2d 101 (Fed. Cir. 1991). To overcome the defense of anticipation, "it is only necessary for the patentee to show some tangible difference between the invention and the prior art." *Del Mar Engineering Lab v. Physio-Tronics, Inc.*, 642 F.2d 1167, 1172, (9th Cir. 1981).

In claim 1 as amended, Applicants recite in the preamble the testing of a "graphical user interface." The Glenn reference refers to the testing of network security protocols and protocol interoperability testing (Glenn et al. pg 148, second column). The protocols tested by the system disclosed in Glenn are for "negotiating security associations" in Cerberus and PlutoPlus. As Glenn states on page 149 in the first column, Cerberus is a stateless IPsec protocol engine that resides in the Linux kernel and PlutoPlus is a key management daemon. Neither of these reference implementations are graphical in nature, they are not themselves graphical user interfaces, nor are their outputs per se graphical objects displayed in a GUI. Conversely, Applicants' claim 1 steps (a), (b) and (c) as amended provide for the selection of test cases for testing the functionality of objects in a GUI, which is not taught by Glenn et al.

Claim 1 step (d), as amended, recites the transmitting of an XML encoded test case file from an application server to a host machine. The Glenn reference teaches HTML encoded test cases, which is a different markup language. A person having ordinary skill in the art distinguishes HTML from XML as different mark-up languages with wholly different tags and tag types. Therefore the claim as amended distinguishes from the cited Glenn et al. reference. (See above, *Scripps Clinic* 927 F.2d 1565).

Claim 1 step (e) as amended, recites that test cases include GUI object functionality steps that are parsed out into a test script. Although Glenn discloses the use of the Perl language to parse code into a series of executable commands, it does so only

with test cases written in HTML rather than XML, as is recited in the amended claim. Furthermore, a person having ordinary skill in the art could not use the same Perl script to parse the XML encoded test case commands as to parse out HTML encoded test case commands. A different script that interprets XML code would have to be written.

Claim 1 step (f) as amended recites that a GUI is to be tested. Applicants respectfully disagree with the Examiner's contention that the elements of step (f) are found in Glenn et al., since Glenn et al. does not disclose an "automation tool GUI environment file." While the "state files" taught in Glen contain control parameters such as "tester configuration variables and IUT addresses" these parameters are mainly used for authentication and security of packets, key exchanges and user information for logging in and transferring data between networks; and mainly Unix run network systems. In contrast, the automation tool GUI environment file includes a wholly different set of parameters that educates automation testing tools (e.g., Winrunner, Visual Test and other equivalents) on the form and function of objects in a graphical user interface so that the automation tool can test the functionality and operability of an interface designed with the same objects.

Claim 12 as amended includes as part of the claimed system that the host machines include computer readable program code for parsing an XML encoded test case into a test script having commands for testing a GUI. As earlier argued, the computer system disclosed in Glenn in Figures 1 and 2 does not teach either of these elements thereby distinguishing this claim from the prior art.

Applicants contend that the rejection of dependent claims 5 and 15, are mooted by the amendments made to the independent claims 1, 13 and 12 from which they respectively depend. Since there is no teaching in Glenn on XML encoded test cases and no disclosure of GUI interfaces or test scripts used to test the objects therein, Applicant respectfully contends that the claims as amended are novel.

Claims 8 and 16 have been amended to include the recitation of XML encoded test cases and the inclusion of steps for testing GUI object functionality. These elements are not taught by Glenn.

The elements of claims 9, as amended and claim 17 are not taught by Glenn et al. Glenn's recitation of a "control and diagnostic interface to the test system through [a] GUI interface tool" is not applicable to claims 9 and 17. The nature of the present invention is the testing of the functionality of objects in a GUI interface using a web based test case management tool (which itself is a GUI). Glenn at page 147, right column, lines 12-16 and page 148, right column, lines 4-20 teach the testing of a Cerberus and PlutoPlus system rather than of a GUI interface, as does the present invention.

Claim 11 is not taught by Glenn because there is teaching of a GUI object or GUI interface to be tested. Glenn's test cases and test system pertain to UNIX based key exchanges and security protocols not GUI objects.

As claims 13 and 19 depend from claim 12 which have been amended to include XML encoded test cases and the testing of GUI objects, similar arguments in support of novelty apply.

Since claims 2, 3, 4, 10, 14 and 18 are canceled; Applicants respectfully contend that their rejection is now rendered moot.

Applicant respectfully requests withdrawal of the Examiner's rejection under 35 U.S.C. § 102.

§103 Rejection of the Claims

To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation to modify the reference or to combine reference teachings. Second, the reference(s) must teach or suggest all the claim limitations. Finally, there must be a reasonable expectation of success. The teaching or suggestion to make the claimed modification and the reasonable expectation of success must both be found in the prior art, not in applicant's disclosure. *In re Vaeck*, 20 USPQ2d 1438 (Fed. Cir. 1991). See MPEP § 2143. If the cited documents do not teach or suggest all the claim limitations, the rejection is improper.

Claim 20 is amended to include the recitation of GUI object test steps and XML encoded test cases. These claim elements are therefore also included by dependency in

claims 21 and 22. Since the Examiner combines Glenn with Hogan to reject the claims on the basis of obviousness, Applicants now contend that the claims as amended include elements not found in the prior art. Accordingly, the combination of Glenn and Hogan do not together teach all of the elements of claims 20-22. Instead, Hogan reiterates all of the elements of the Cerberus and PlutoPlus for their uses with IPsec and their testing using the NIST IPsec-WIT system. Accordingly, Hogan does not teach the testing of a GUI, XML encoded test cases or test scripts that include syntax that tests the functionality of GUI objects.

Accordingly, Applicants submit that the combination of Glenn and Hogan do not disclose or teach the invention and requests withdrawal of this rejection of claims 20-22 under 35 U.S.C. § 103(a).

CONCLUSION

Applicant respectfully submits that the claims are in a condition for allowance, and notification to that effect is earnestly requested. The Examiner is invited to telephone Applicants at (510) 625-3127 to facilitate prosecution of this application.

Respectfully submitted,

KONSTANTIN MELAMED

YEVSEY MELAMED

Date 08/11/2006

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail, in an envelope addressed to: Commissioner of Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on this 11 day of August, 2006.

Konstantin Melamed
Name

Signature

Yevsey Melamed
Name

Signature